

**NASA**  
**Technical**  
**Paper**  
**2695**

February 1987

# A General Solution to the Silhouette Problem

David R. Hedgley, Jr.

**NASA**

**NASA  
Technical  
Paper  
2695**

1987

# A General Solution to the Silhouette Problem

David R. Hedgley, Jr.

*Ames Research Center  
Dryden Flight Research Facility  
Edwards, California*



National Aeronautics  
and Space Administration

Scientific and Technical  
Information Branch

## INTRODUCTION

The requirement for flexibility in choosing how to display computer-generated three-dimensional drawings has become increasingly important in recent years. One major consideration is the enhancement of the realism and aesthetics of the presentation. A polygonal representation of objects, even with hidden lines removed, is not always desirable. A more pleasing pictorial representation often can be achieved by removing some of the remaining visible lines (fig. 1). Consequently, it would be advantageous to have the facility to render any subset of polygons as a silhouette with respect to itself. Furthermore, the vocabulary of objects should have no restrictions (that is, the solution should be flexible enough to apply to every class of objects).

While the silhouette problem has been addressed before, there has been no completely general solution. Moreover, the known partial solutions (unpublished) require so much topological information that data preparation is extremely difficult. This report discusses the first completely general solution to this most classical problem and describes the implementation of the solution.

The author wishes to express his appreciation to Ralph Carmichael of NASA Ames Research Center for his many helpful suggestions and constructive criticisms.

## DISCUSSION

The silhouette solution presented in this report is based on the following definitions.

1. Define  $X'$  to be the projection of  $X$ .
2. A boundary point of a set  $S$  is a point  $(x, y, z)$  whose projection  $(x, y)$  has the property that every neighborhood lying in the projection plane and containing  $(x, y)$  intersects  $S'$  and its complement.
3. A boundary line is a line composed of boundary points.
4. Relevant polygons with respect to a given polygon  $A$  are those polygons whose projections must be considered in determining if a line of  $A$  is a boundary line.
5. A silhouette is a rendering that is a composite of the projections of all boundary lines.
6. An interior point  $(x, y, z)$  is a point whose projection  $(x, y)$  has the property that there exists a neighborhood lying in the projection plane and containing  $(x, y)$  that itself is completely contained in the interior of a group of polygons. Note that a point  $P'$  lying on the edge of a polygon  $A'$  is defined to be contained by  $A'$ . (See ref. 1, pp. 3 and 4, for the criterion used to determine whether a point  $P'$  is contained by a given polygon  $A'$ .)

Using these definitions, we shall now state and prove a theorem that addresses the process of selecting points on an arbitrary line. Furthermore, this selection of points will be sufficient to determine a silhouette for an arbitrary solid.

**THEOREM** *Let  $L$  be any line belonging to any three-dimensional scene  $S$ . Also, let  $P_1, \dots, P_n$  be all the ordered points (see ref. 1, p. 5) on  $L$  whose projections represent intersections of  $L'$  with the lines  $L'_1, \dots, L'_n$ . If a segment  $L_S$  of  $L$ , determined by  $P_i$  and  $P_{i+1}$ , is visible, then every interior point of  $L_S$  has the same character; that is, every interior point of  $L_S$  is a boundary point or not one of them is.*

**PROOF** The visibility of  $L_S$  can be determined with a hidden-line algorithm. If  $L_S$  is invisible in the hidden-line sense, it is clearly not a boundary line in the silhouette sense. Therefore, the status of  $L_S$  must be considered only if it is visible.

Let  $P_m$  be a boundary point that is an interior point of  $L_S$ . Also, let  $P_k$  be any other interior point of  $L_S$ . The theorem asserts that  $P_k$  is also a boundary point. Assume the contrary. Then  $P_k$  is an interior point of some polygon  $A$ , and thus  $P'_k$  lies in the interior of the projection of a polygon  $A$  or on the projection of its edge. Now if  $P'_k$  belongs to the interior of a polygon, say  $A'$ , it follows that  $L'_S$  does not belong to any edge of  $A'$ . However,  $P_m$  is a boundary point, and thus  $P'_m$  must lie on an edge of  $A'$  or outside  $A'$ . In either case, there would be an intersection on  $L'_S$  between  $P'_i$  and  $P'_{i+1}$ , which is impossible. On the other hand, if  $P'_k$  belongs to an edge of  $A'$ , then there is a subset of  $L'_S$ , say  $L'_r$ , that belongs to at least two polygons; otherwise,  $P'_k$  would not be an interior point. But again,  $P_m$  is a boundary point, and hence  $P'_m$  cannot belong to  $L'_r$ ; that is,  $P'_m$  is outside  $A'$ , while  $P'_k$  is contained by  $A'$ . Therefore, since the polygons are closed, it follows that there is an intersection on  $L'_S$  between  $P'_k$  and  $P'_m$ , and hence between  $P'_i$  and  $P'_{i+1}$ , which is a contradiction. The argument assuming that  $P_m$  is not a boundary point is similarly developed.

By extension, one can define a silhouette subset to be the silhouette with respect to a proper subset of  $S$  only. This flexibility allows any grouping of polygons to be a family (see fig. 2) and therefore permits any combination of silhouette and nonsilhouette specifications (see fig. 3). Further, the solution is able to deduce segments of lines that do not contribute to a silhouette (Figs. 2 and 4).

#### IMPLEMENTATION

An algorithm was written based upon the concepts of reference 1 and the theorem presented in this report.

While the theorem addresses the questions of economy and generality, the real issue of algorithmic efficiency is ultimately predicated upon the initial size of the relevant set of polygons for a given polygon. (The fineness of the neighborhood can also be a contributing factor, but it is not of major concern.) It should be noted, however, that the relevant set in the silhouette case is larger, on the average, than in the hidden-line case because polygons "behind" the given element are also relevant.

Initial relevance is determined as in reference 1, with the exception that the logarithmic search for z values is not performed. This means that the performance of the silhouette algorithm as written is only a little slower than that of the hidden-line algorithm of reference 1; it is also linear. However, it should be noted that the theorem is applicable to any hidden-line algorithm.

Moreover, the warped-polygon solution is a free corollary. Any triangular subdivisions of a polygon can be regarded as members of the same family; hence, the rendering of its silhouette is a polygon with no interior triangles.

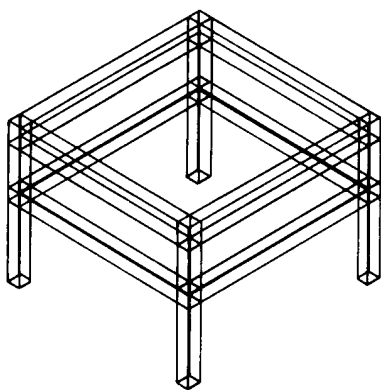
#### CONCLUDING REMARKS

In this report, the silhouette problem for calligraphic drawings is solved for the most general case. This solution offers all possible combinations of silhouette and nonsilhouette specifications for an arbitrary solid. It therefore allows the flexibility to enhance the clarity of any three-dimensional scene presented in two dimensions.

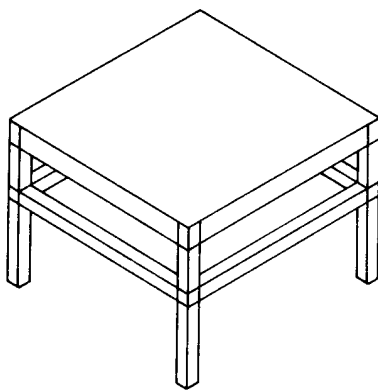
*National Aeronautics and Space Administration  
Ames Research Center  
Dryden Flight Research Facility  
Edwards, California, April 16, 1986*

#### REFERENCE

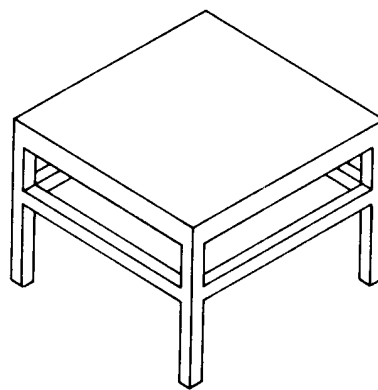
1. Hedgley, David R., Jr.: A General Solution to the Hidden-Line Problem. NASA RP-1085, 1982.



*All lines shown*

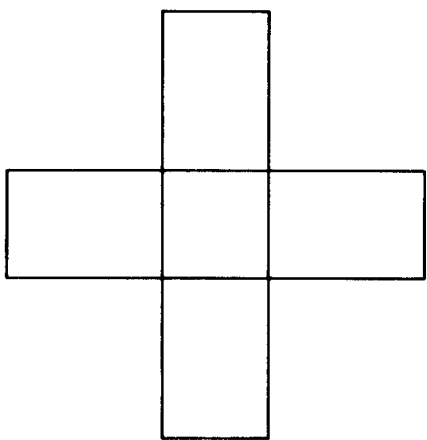


*Hidden lines removed*

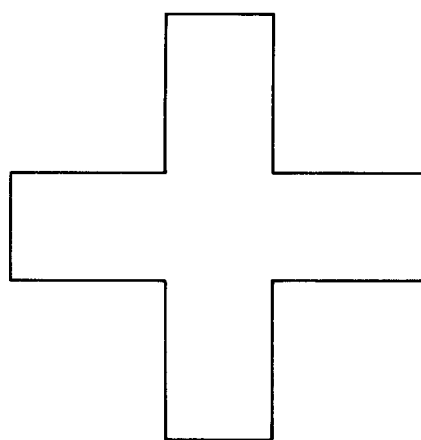


*Some visible lines removed*

*Figure 1. Types of computer-generated drawings.*

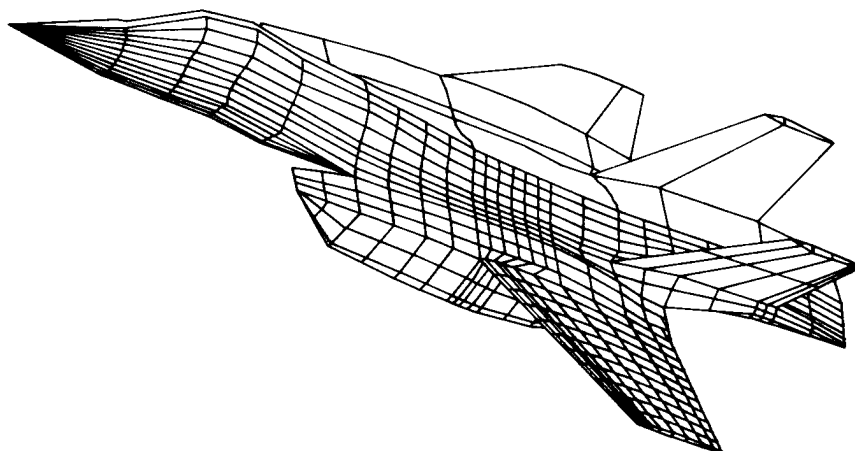


*No grouping*

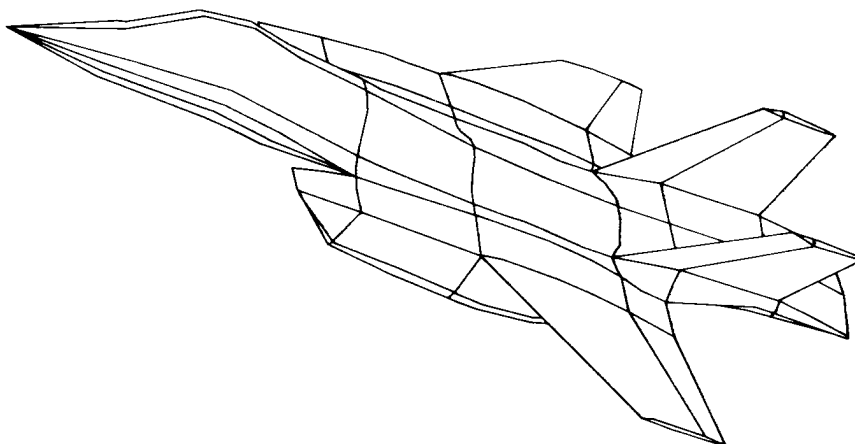


*Family of two polygons*

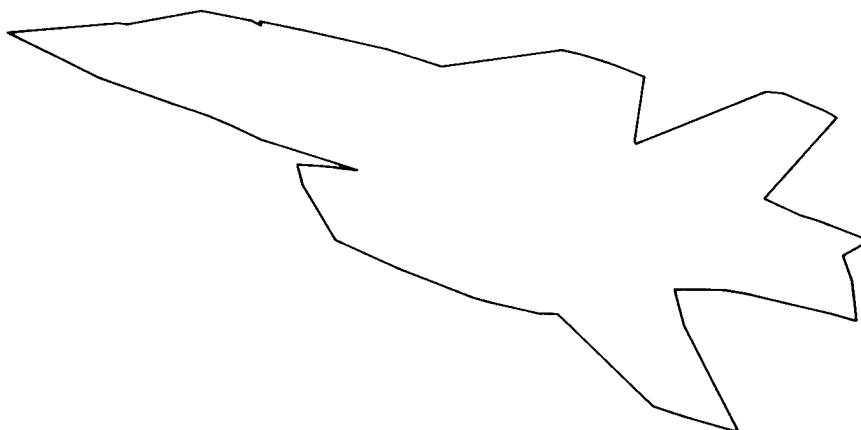
*Figure 2. Groupings of polygons.*



*Combination of specified families and nonfamilies*

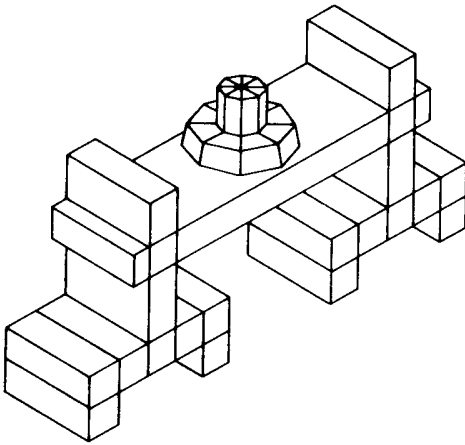


*Several families*

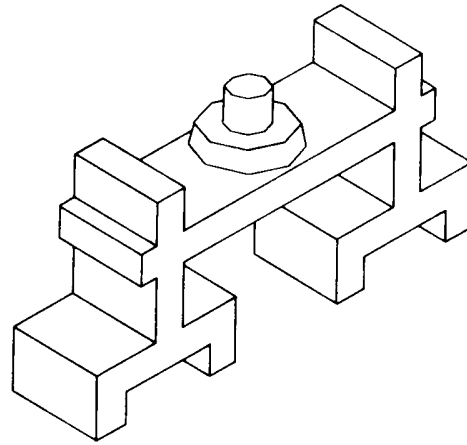


*One family*

*Figure 3. Specifying families of polygons.*



*No grouping*



*Many families*

**Figure 4. Removing line segments not contributing to a silhouette.**

